RESEARCH PROPOSAL

Design and Development of Framework for Improvising Software Testing in Cloud Environment

For registration to the Degree of
Doctor of Philosophy

IN THE FACULTY OF SCIENCE

THE IIS UNIVERSITY, JAIPUR

Submitted By:

Pushpanjali Saini

ICG/2017/24626

Under the Supervision of:

Dr. Shveta Parnami

Associate Professor

Department of Computer Science & IT

August, 2018

Design and Development of Framework for Improvising Software Testing in Cloud Environment

1. Introduction:

Software Testing is one of the important phases of Software Development Life Cycle. Testing a software demands dedicated infrastructure and resources for analyzing application's speed, performance, security, etc. Moreover, growing complexity of applications demand flexible real time environment which is difficult to maintain within premises. To address this issue, Cloud computing is an effective approach for providing enhanced performance, security, efficiency and minimal cost requirements (Vijay and Hariharan(2016)).Testing in the cloud influences cloud computing environment and seek to simulate real-world user traffic. The cloud-testing also provides availability of unlimited resources at pay per use concept (Jun et. al., (2011)).

1.1 Cloud Computing

Cloud Computing refers to manipulating, configuring, and accessing the hardware and software resources remotely. It offers online data storage, infrastructure and application. It is an internet dependent technology which provides the services to user, small and large organizations, on demand. Cloud computing stores the user data and maintain data center of cloud provider like Amazon, Oracle, and Google, Microsoft etc. (Pund and Dandge (2017)).

According to the official NIST definition, "cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

1.2 Software Testing

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Testing is the ongoing activity for evaluating the functionality of the system with the objective to find whether it meets the specified requirements or not. In simple words, testing is executing a system in order to locate bugs, errors, or missing requirements in contrary to the actual requirements before delivering it to customer. It also helps in detecting unwanted system behavior and also verifies and validate that the software works properly.

According to ANSI/IEEE 1059 standard, Testing can be defined as "A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item".

1.3 Need for Cloud Based Software Testing

Testing software requires enough resources and budget to complete it successfully. But most of the organizations face the challenges to provide sufficient resources to test their software in distributed environment, with different loading level. This leads to severe problem when the software is deployed into different client environment and with varying user load. Cloud computing is a one of the emerging technology which opens new door for software testing (Katherine and Alagarsamy (2012)). Cloud

computing is opening up new vistas of opportunity for testing. Testing has traditionally been viewed as a necessary evil because it required a huge, dedicated infrastructure and resources that were used sporadically ('Cognizant Report' 2011). Cloud computing not only changes the way of obtaining computing resources (such as computers, infrastructures, data storage, and application services), but also changes the way of managing and delivering computing services, technologies, and solutions (Gao et. al., (2011)).

Deploying and migrating software applications to the cloud involves many combinations of options that vary widely in their characteristics and performances, from different combinations of CPU, memory, storage, and network options to IT resource management services, to algorithms that can perform dynamic resource scaling (Almory et. al., (2016)). However, the technological virtualization has met base-level operational and financial objectives by eliminating the need for intensive capital investments. Cloud-based testing has the potential to offer a compelling combination of lower costs, pay-per-use and elimination of upfront capital expenditures i.e Cap-Ex('Cognizant Report' 2011).The software testing on cloud lowers the cost of acquiring infrastructure, saves time consumed in evaluation and makes proper resource utilization (Almory et. al., (2016)). At the same time, cloud-based testing introduces set of challenges, such as data security and a lack of standards, especially in the public cloud model ('Cognizant Report' 2011).

1.4 Software Testing Techniques in Cloud Environment

Various testing techniques are used for effective software testing. In context to the cloud environment below mentioned are frequently used techniques which enhance software testing in cloud environment and outshine other testing techniques of the same categories.

Prioritization is the activity that arranges items or activities in order of importance relative to each other. The technique is developed in order to run test cases of higher priority in order to minimize time, cost and effort during software testing phase. Prioritizing test cases can be done on the basis of requirements, costs of bug fixing, history of the parent device etc. There are numerous methods on how to prioritize requirements such as Ranking, Voting, Numerical assignment MoScoW, Bubble sort, Hundred dollar, Dependency map etc.

Clustering or Cluster analysis is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense) to each other than to those in other groups (clusters). The objective of any clustering algorithm is to sort entities into groups, so that the variation between clusters is maximized relative to variation within clusters. It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, and computer graphics.

The different types of clustering methods include Partitioning methods, Agglomerative Hierarchical clustering, Fuzzy clustering, Density-based clustering and Model-based clustering.

Mutation Testing is a type of software testing in which certain statements in the source code are mutated (changed) in order to check whether test cases are able to find the errors or not. The mutations introduced to source code are designed to emulate common programming errors. The objective of Mutation Testing is to assess the quality of the test cases which should be strong enough to fail mutant code.

Mutation testing could be fundamentally categorized into 3 types– statement mutation, decision mutation, and value mutation

Fault injection is software testing technique by introducing faults into the code for improving the coverage and usually used with stress testing for robustness of the developed software. The objective is to evaluate the software behavior in terms of its ability to detect faults. Fault injection is categorized into 2 types - Hardware fault injection and Software fault injection

Hardware fault injection uses additional hardware to introduce faults into the target system's hardware. Depending on the faults and their locations, hardware-implemented fault injection methods fall into two categories:

- Hardware fault injection with contact
- Hardware fault injection without contact

Software fault injection techniques are attractive because they don't require expensive hardware. Furthermore, they can be used to target applications and operating systems, which is difficult to do with hardware fault injection.

- Compile-Time Injections
- Run-Time Injections

The tools used for Software Fault Injection are BStorm, The Mu Service Analyzer, Holodeck and Xception

2. Review of Literature:

In order to find the relevancy and utility of moving testing from local systems to cloud environment multitude of research papers were referred and studied.

Garg and Lavingia (2017) analyzed the impact of cloud computing on conventional software engineering. The study also stated challenges such as Mobility, Multitenancy, Parallelization, etc. that are being faced by conventional models to test the software. The author also proposed Triple-I model to mitigate these challenges and reap the benefits of advanced technology.

Tiitinen (2013) presents information on software testing, its importance, process to select tools for testing and outcomes in case if serious defects occurs in software. The software testing tool is key element to conduct successful testing. In this study seven classifications of software testing tools are included which are categorized according to ISTQB classification. Most popular software testing tools and their usage are presented and cloud service tools are also considered. Since the testing tool selection process is complicated so, information is specified from the traditional selection process for software testing tools and test automation tools.

Incki et. al., (2012) provides an overview regarding main contributions, trends, gaps, opportunities, challenges and possible research directions of testing a software in cloud environment. This paper also differentiated between two concepts 'Testing for the cloud' and 'Testing on the cloud'. 'Testing for the cloud' refers to testing of applications that are specifically developed for cloud platform and 'Testing on the cloud' provides testing of applications that reside either on-premises or on the cloud but deployed on platform other than cloud. The research also evaluated that acceptance testing, interoperability testing and test task management are open research areas for testing over the cloud.

Mungekar and Toradmalle (2015) paper identified main testing types such as functional testing, unit testing, load testing, stress testing, performance testing, security testing etc. that are provided by cloud service providers and also discussed the benefits of cloud based testing. This paper also provided cloud testing life cycle from generation of test scenarios to the delivery of result as shown in figure 1:-

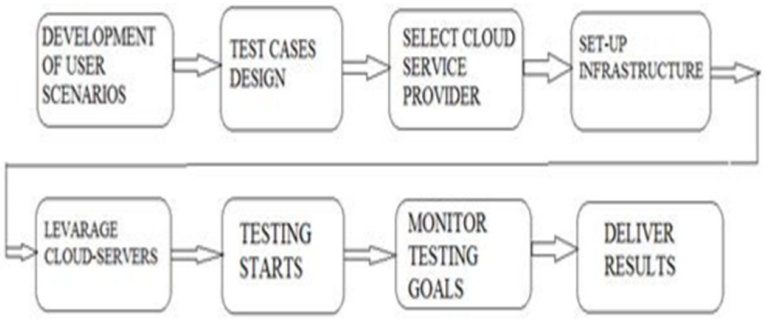


Fig 1: Cloud testing life cycle (Mungekar, S. D., and Toradmalle, D. K.., 2015)

Above cloud testing life cycle starts with development of scenarios followed by design of test cases, selection of cloud service provider, infrastructure set up, leverage cloud servers, preform testing, monitor testing objectives and at the end delivers output.

Jun et. al., (2011) described the need of performing software testing on the cloud which includes: 1) cost saving, 2) improved testing efficiency, 3) performance testing is more realistic and 4) changes in external environment. The author also attempted to answer the question to which software testing project can perform testing on cloud, why to do cloud testing and how to do cloud testing? It also revealed the fact that information security is also major concern in cloud based testing.

Gao et. al., (2011) provided a comparative view between conventional software testing and cloud based software testing. The author also discuss issues such as on-demand test environment, scalability and performance testing, testing security in cloud, on demand testing challenges, Integration and regression testing issues. The study revealed four needs in cloud based testing which includes: 1) adequate test models and criteria, 2) TaaS test process and QoS standards, 3) innovative test methods and solutions and 4) innovative dynamic test platforms and tools.

Siddiqui and Ahmad (2016) has discussed four software testing approaches for testing approaches for testing framework in cloud computing.

(i) Prioritization technique is used to run and develop test cases in which priorities are assigned based on cost, test case dependencies, complexity and customer priority.

(ii) Clustering is the statistical technique used for analysis of data. The Agglomerative clustering is the bottom up method which uses squared error objective function for assembling data sets and Priority based clustering is used to design algorithms by using part of data.

(iii) Load Balancing is used to avoid overloading of resources and eliminate response time while extending throughput. The Static LB, Hierarchal LB and Centralized LB techniques are used to arrange nodes

(iv) In security mechanism Diffie Hellman algorithm (DHA), Data Encryption Standard (DES) and Advanced Encryption Standard (AES) Algorithms are used for encryption and decryption of data.

The paper concluded that Dependency Structure Prioritization (DSP) is better than other priority techniques in terms of cost reduction, in Clustering Agglomerative Clustering (AC) algorithm takes less computational time, the Hierarchal Load Balancing (HCB) technique shows better

results in terms of resource utilization and the DH algorithm considered better that other security mechanisms.

Vijay and Hariharan (2016) has proposed software testing framework on weight based prioritization technique in distributed cloud environment. The cosine similarity is used to determine weight of similar test cases and K-Medoid clustering is used for similar test cases. Moreover, key based Elliptic Curve Cryptography (ECC) algorithm is used to provide security. The author stated that experimental results proved proposed framework is better that existing techniques and algorithms.

Riungu et. al., (2010) identified research issues that arises in testing software using cloud computing services. These were gathered by interviewing industry practitioners from eleven software organizations. By comparing the findings with existing literature these issues were categorized as Application, Management, Legal and Financial issues. The authors introduced the grounded theory which is a set of research procedures leading to the emergence of conceptual categories. It is an approach used for creating theories from collected and analyzed data. (Seaman C. B., (1999)

Liu et. al., (2013) had summarized different products and solution of cloud testing and its environment i.e TEaaS. The author stated that before migrating test to cloud user should consider testing efficiency, test case independency and environment requirements that affects the testing execution. Moreover, this paper provided practical solutions to cloud testing such as D-cloud developed by [Banzai et al., (2010)] with fault injection facility and testing system developed by [Lian yu et al., (2010)] based on task clustering.

Riungu et. al., (2010) identified six testing services already being deployed on cloud including: D-Cloud, Cloud 9, YETI, ASI, VATS and RNL.

Banzai et. al., (2010) from University of Tsukuba in Japan developed 'D-Cloud', the software test environment for parallel and distributed systems. It is built upon Eucalyptus, an open source cloud and uses QEMU, an open source virtual machine for fault injection. It facilitates Fault tolerance testing and governs virtual machines with fault injection facility.

Cloud 9 is a web service that initiates execution of computer clusters operating on public cloud infrastructures such as Amazon EC2 and also on clusters that execute cloud software such as Eucalyptus. The cost of service depends upon the program size. (Ciortea, et al. (2010))

YETI i.e York Extensible Testing infrastructure is an automated testing tool with ability to test programs written in different languages. YETI inhibits performance and security issues which could be resolved by migrating test to cloud and secure data by executing tests on clean virtual machines. (Oriol and Ullah, (2010))

AST i.e Autonomic Self Testing service that uses processing capabilities of cloud to enhance self-testing process (King and Ganti, (2010))

VATS i.e Virtualization-Aware Automated Testing service which enhance testing by executing test and managing virtual infrastructure. (Gaisbauer., et al. (2008))

RNL i.e Remote Network Labs are used to build on-demand virtual test labs. (Liu and Orban, (2010))

Besides these, Jata has been created to support distributed component testing. It is a Java framework which includes a meta-model for defining and structuring test cases. The meta-model uses concepts from the standardized testing language TTCN-3 (Testing and Test Control Notation version 3). (Hlodver Tomasson, (2011))

The above review of literature shows that testing is major phase in software development life cycle. If testing is conducted within the cloud environment it would reap series of welfares in terms of cost, time performance, efficiency, etc.

3. Motivation/Justification and Relevance:

In the light of the literature reviewed, it is observed that testing plays a vital role in software development. With the advancement of technology, cloud services are heavily benefiting software testing process and is more advantageous comparative to on-premises testing or conventional methods of testing. The conventional testing requires huge, dedicated infrastructure and resources which becomes too expensive. Moreover, with growing complexity of applications, it is difficult for organizations to maintain real time environment.

So, Cloud based testing is efficient way to enhance software testing process. The technological virtualization has the ability to meet operational and financial objectives. Cloud based testing lowers the cost by providing pay per use facility. Moreover, it also provides non-cost benefits such as on-demand flexibility, enhanced collaboration, freedom from holding assets, greater level of efficiency and saves time. Such factors increased the demand for cloud based testing.

Migrating testing on cloud also presents set of challenges such as lack of resources, real time environment, proper communication and inefficient test coverage. Addition to these testing on the cloud also introduces limitations i.e Service Level Agreements, limited infrastructure, data security, confidentiality, integrity and lack of standards especially in public cloud. According to Cognizant Report "Moving testing to cloud should be viewed as strategic initiative rather than tactical objective.

The challenges and gaps in reviewed literature while taking testing to cloud has encouraged to develop the new framework or modify the existing one for effective software testing. It is required to select and simulate suitable cloud infrastructure/ platform as test bed that are equipped with diverse and scalable computing resources, system infrastructures and tools for testing techniques which is prioritization, clustering, mutation and fault injection.

4. Objectives:

The objectives of the present research are:

1) To make a detailed study of software testing strategies such as Prioritization, Clustering, Fault Injection and Mutation Testing for testing applications on cloud environment.

2) To study and analyze the existing cloud test environment strategies and testing tools (Research based/ commercial) provided by different cloud service providers for software testing strategies.

3) To generate new or improvise the existing cloud framework for software testing based on testing algorithms/technique and simulate on cloud environment.

4) To compare the efficiency and effectiveness of the improvised / the newly developed system with existing software testing techniques on clouds.

5. Methodology:

In order to achieve the above mentioned objectives the following methodology will be followed:

1) To study, analyze and explore the software testing strategies (Prioritization/ Clustering/ Fault Injection/ Mutation Testing) on cloud environment, we propose to study related research papers.

2) Evaluate the effectiveness of existing test environments such as Cloud 9, D-cloud, YETI, ASI, VATS, RNL and Jata test framework.

3) Evaluate the performance of identified test environments on the basis of cost, time, security, efficiency and other performance related parameters.

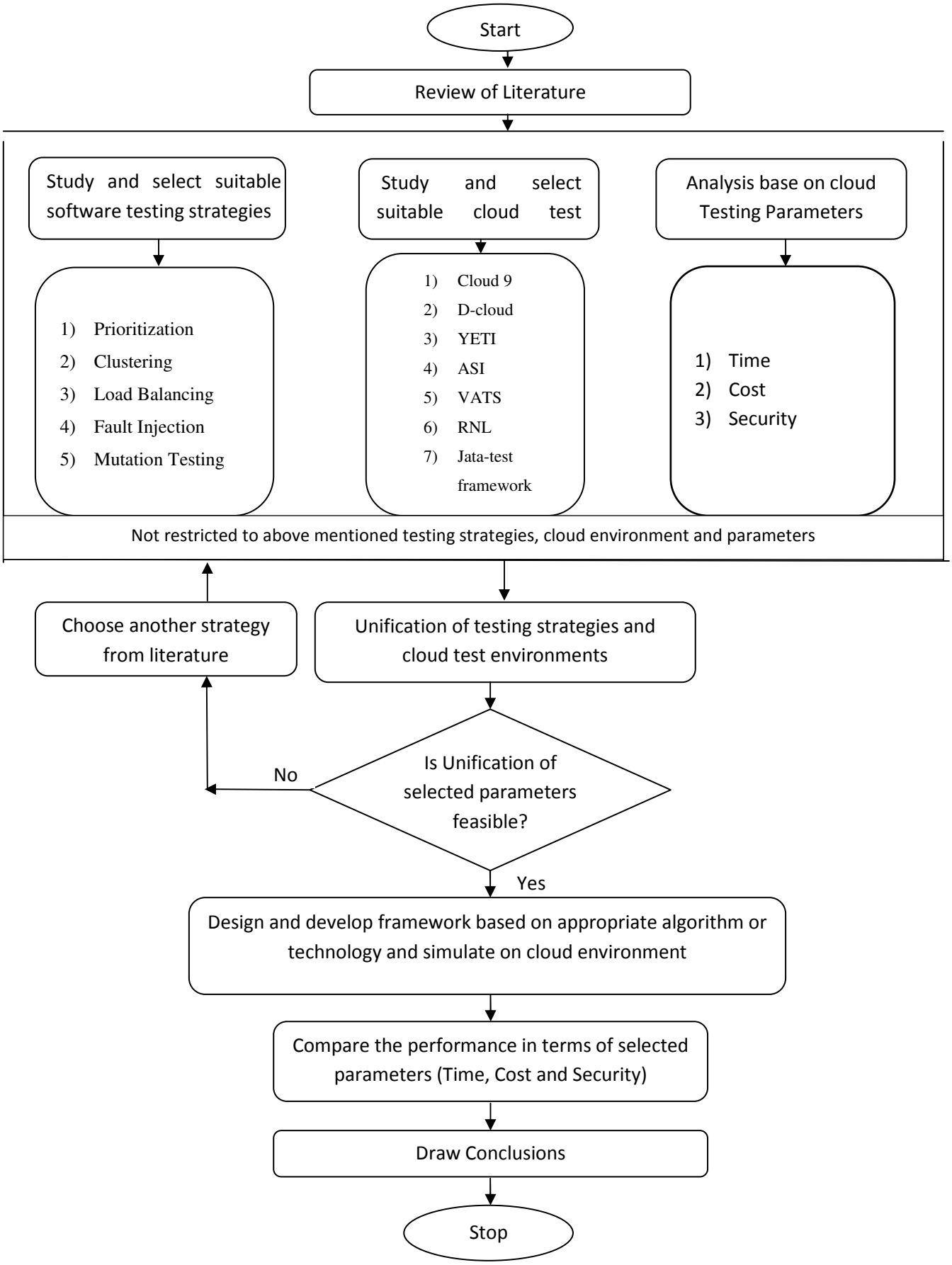
4) Design and develop new/refine existing test framework for deploying software testing on cloud by the unification of selected test environment and testing strategies and to simulate the cloud environment to record the performance of testing strategies.

5) Compare the performance and effectiveness of newly designed framework with existing one.

6) On the basis of comparison the conclusions will be drawn.

7) Presentation of new results in the form of papers and thesis.

6. Flow Chart:


Start
Review of Literature
Study and select suitable software testing strategies
Study and select suitable cloud test
Analysis base on cloud Testing Parameters
1) Prioritization
2) Clustering
3) Load Balancing
4) Fault Injection
5) Mutation Testing
1) Cloud 9
2) D-cloud
3) YETI
4) ASI
5) VATS
6) RNL
7) Jata-test framework
1) Time
2) Cost
3) Security
Not restricted to above mentioned testing strategies, cloud environment and parameters
Choose another strategy from literature
Unification of testing strategies and cloud test environments
No
Is Unification of selected parameters feasible?
Yes
Design and develop framework based on appropriate algorithm or technology and simulate on cloud environment
Compare the performance in terms of selected parameters (Time, Cost and Security)
Draw Conclusions
Stop

7. Tools and Techniques:

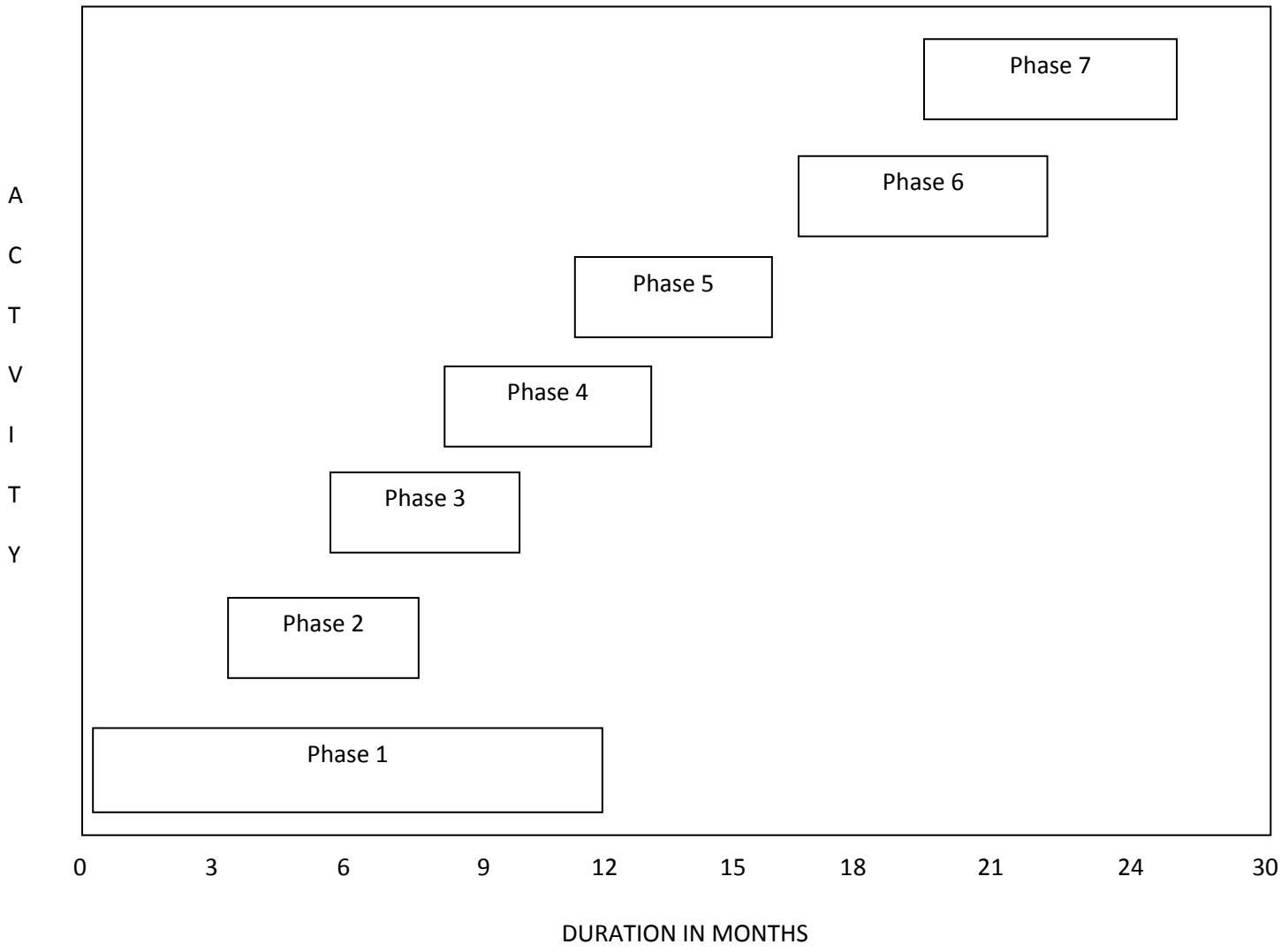
Several research based testing services are deployed on cloud such as cloud 9, D-Cloud, YETI, AST, VATS, RNL, etc. The proposed framework will be compared with the existing services with respect to performance parameters. Other commercial tools such as SOASTA, LOAD STROM, BLAZEMETER, NESSUS, etc. could be used as per requirements of the proposed research based on the type of testing to be performed.

8. Work Plan:

The research plan is divided into following seven phases:

Phase	Activity	Approximate time (in months)
1.	Literature survey on software testing through cloud computing	8-12
2.	Identification of software testing strategies on cloud	3-4
3.	Assessment of the effect of the unification of different strategies with existing framework or testing services	3-4
4.	Analyze existing testing tools (Research based/ Commercial) provided by different cloud service providers	3-4
5.	Develop new / refine existing framework for software testing based on some particular algorithm or technique.	3-4
6.	Implementation, comparison with existing techniques and drawing conclusions	4-5
7.	Paper/ Thesis Writing	4-6

The timeline chart for research plan is as follow:



9. References:

Almorsy, M., Grundy, J., and Müller, I (2016). An analysis of the cloud computing security problem. *arXiv preprint.* **arXiv:1609.0110.**

Anitha, D., and Srinath, M. V (2014). A review on software testing framework in cloud computing. *International Journal of Computer Science and Information Technologies*. ***5*, 7553-7562.**

Banzai, T., Koizumi, H., Kanbayashi, R., Imada, T., Hanawa, T. and Sato, M (2010, May). D-cloud: Design of a software testing environment for reliable distributed systems using cloud computing technology. *IEEE/ACM International Conference on* Cluster*, Cloud and Grid Computing (CCGrid),* **631-636.**

Seaman C.B., (1999) "Qualitative methods in empirical studies of software engineering*." IEEE Transactions on Software Engineering.* **vol. 25**, pp. 557-572.

Gao, J., Bai, X. and Tsai W. T., (2011). Cloud testing-issues, challenges, needs and practice. *Software Engineering: An International Journal*. ***1*(1)**, 9-23.

Garg R. and Lavingia K., (2017)**,** Survey Paper on Impact Of Cloud Computing on Conventional Software Engineering**.** *International Journal on Computer Science and Engineering.* **Vol. 9 No.05, ISSN: 0975-3397**

Grundy, J., Kaefer, G., Keong, J. and Liu, A. (2012). Guest editors' introduction: Software engineering for the cloud. *IEEE software*. ***29*(2), 26-29.**

H. Liu and Orban D., (2010, January) "Remote network labs: An on-demand network cloud for configuration testing." *ACM SIGCOMM Computer Communication Review*. **vol. 40,** num.

Incki, K., Ari, I. and Sözer, H. (2012, June). A survey of software testing in the cloud. *IEEE Sixth International Conference on* In *Software Security and Reliability Companion (SERE-C)* (pp. 18-23).

Jun, W. and Meng, F. (2011, September). Software testing based on cloud computing. *IEEE International Conference on Internet Computing & Information Services (ICICIS).* (pp. 176-178).

Katherine A. V. and Alagarsamy K., (2012). Software testing in cloud platform: a survey. *International Journal of computer applications*. *46*(6), 21-25.

Ciortea L., Zamfir C., Bucur S., Chipounov V. and Candea G. , (2009, October) "Cloud9: A software testing service." *In Proc. of the 3rd SOSP workshop on Large Distributed Systems and Middleware (LADIS), Big Sky, MT.*

Liu Y., (2013). Cloud-based Testing: Opportunities and Challenges". *Cloud-Based Software Engineering," **Cloud-Based Software Engineering*, 40**.

Oriol M. and Ullah F., (2010, April) "YETI in the cloud." *In proc. 2nd International Workshop on Software Testing in the cloud (STITC), 3rd IEEE International Conference on Software Testing, Verification and Validation (ICST).*

Mungekar, S. D., Toradmalle and D. K. (2015). Cloud Testing-An Advancement in Software Testing. *International Journal of Application or Innovation in Engineering and Management*, *4*(3), 61-63.

Pund M. A. and Dandge S. S., (2017). An Overview of Cloud Computing: Platforms, Security Issues and Applications. *International Journal, 2(5).*

Riungu L. M., Taipale O. and Smolander K., (2010, November). Research issues for software testing in the cloud. *IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom)* (pp. 557-564).

Banzai T., Hanawa T., Koizumi H., Kanbayashi R., Imada T. and Sato M., (2010, April) "Large-scale software testing environment using cloud computing technology for dependable parallel and distributed systems." *In proc. 2nd International Workshop on Software Testing in the Cloud (STITC), 3rd IEEE International Conference on Software Testing, Verification and Validation (ICST).*

Tiitinen, M. (2013). Key Factors for selecting software testing tools.

King T.M. and Ganti A.S., (2010, April) "Migrating autonomic self-testing to the cloud," *In proc. 2nd International Workshop on Software Testing in the Cloud (STITC), 3rd IEEE International Conference on Software Testing, Verification and Validation (ICST).*

Gaisbauer S., Kirschnick J., Edwards N. and Roila J., (2008) "VATS: virtualized-aware automated test service," *In proc. 5th International Conference on Quantitative Evaluation of Systems*.

Siddiqui, T. and Ahmad, R. (2016). A review on software testing approaches for cloud applications. *Perspectives in Science*, *8*, 689-691.

Vijay, J. F. and Hariharan, B. (2016). Creation of Software Testing Environment in Cloud Platform. *Asian Journal of Information Technology*, *15*(13), 2228-2237.

Technical report by Cognizant 2011, Taking Testing to the Cloud.